

claims must be present in the reference, and if even one limitation is missing from the reference, then it does not anticipate the claim. *Kloster Speedsteel AB v. Crucible, Inc.*, 793 F.2d 1565 (Fed. Cir. 1986). Burgun fails to satisfy this rigorous stand.

As discussed in the background, the present application is concerned with testing a representation of a device/component (sometimes called a device under test (DUT)) prior to the production of an actual device based on that representation. It is possible to make a particular representation of a component configurable, for example in situations where the actual construction of the component may depend on the arrangement of other components within the system with which that component is intended to interact. When using such configurable representations, it is possible to generate many different "instantiations" of the component representation. Until the present invention, it has generally been considered impractical to provide a testbench which can test in isolation the configurable component representation to ensure that a working instantiation of that representation can be created under many or all configuration options. As a result, when a particular instantiation of a configurable component representation is produced, that representation is typically placed within a representation of a portion of the system with which the component representation will interact so that reliable testing can take place. But this approach relinquishes direct control over the input stimuli to the component representation being tested.

To overcome this limitation, the inventor conceived of a different approach. A testbench providing a test environment for a representation of a device is generated. Because the device representation is configurable based on configuration data, the testbench is generated with reference to the same configuration data and a first set of templates defining the test environment. The testbench then matches the top-level of the device representation thereby

allowing many possible device configurations to be tested once that particular device configuration is generated. In the words of the specification, a matching testbench is generated for any particular device instantiation, which allows it to be tested in isolation, and thus, permits direct control of the input stimuli to that device instantiation. After a device representation and its matching testbench have been so generated, a simulation tool runs a model of the data processing apparatus using the device representation and the testbench to generate test results which can be analyzed to determine whether the device representation is operating as desired.

Burgun describes emulating a design under test associated with a test environment. Reconfigurable hardware systems, for example FPGA circuits, may be used to perform such emulation. The part of the test environment embedded in the reconfigurable hardware system and the design under test are both typically implemented by the same configurable resources, and this consequently results in a “very strong dependence between the test environment and the design under test.” See paragraph 17. Burgun’s solution to this problem is to provide a simulation/emulation platform in which the reconfigurable system emulating the design under test is separated from that emulating the hardware part of the test environment. See paragraphs 23-25.

But there is a fundamental difference between Burgun and the pending independent claims: Burgun assumes that there is already a design under test and an associated test environment. Working from that assumption, Burgun configures reconfigurable hardware circuits to simulate the operation of that design under test and associated test environment. Thus, Burgun is directed to the simulation process described in the background section of the application on page 6, lines 12-16. In contrast, claims 1, 23-25, and 46 are directed to the earlier

step of producing a representation of the device under test and its associated test environment, so that they can subsequently be simulated using a simulation tool.

The Examiner relies on paragraphs 28 and 32 of Burgun. Paragraphs 26-30 describe emulating a design under test associated with a test environment (see paragraph 26). This process involves generating two different configuration files. The first configuration file is used to configure a first reconfigurable hardware part used to represent a test bench (paragraph 28), and the second configuration file is used to configure a second reconfigurable hardware part used to emulate the design under test (paragraph 29). As stressed in paragraph 29, the first and second reconfigurable hardware parts are “distinct and mutually connected.” These two separate parts are provided with their own separate configuration files to overcome the problems associated with the “strong dependence” between the test environment and the design under test.” See paragraph 17.

Accordingly, Burgun’s configuration files are not used to “configure the representation of the device” and to generate the associated “testbench with reference to the configuration data and a first set of templates defining the test environment.” Instead, Burgun’s configuration files are used to configure the reconfigurable hardware units used to emulate an already-existing design under test and associated test environment.

Burgun also stresses using two different configuration files: the first configuration file to configure the reconfigurable hardware representing the testbench and the second configuration file to configure the reconfigurable hardware unit acting as an emulator for the device under test. In contrast, claim 1 recites at step (b) the testbench is generated with reference to the configuration data and a first set of templates. From the earlier recitations in claim 1, the

configuration data being referred to is the same configuration data used to configure the representation of the device.

The Examiner also refers to paragraph 32 in relation to step (b) of claim 1. Is the Examiner taking the view that the compilation directives referred to therein are in some way analogous to the claimed first set of templates? Such a position is inconsistent with Burgun's teachings in paragraph 32 that the result of the generating step which makes use of the compilation directives is the first configuration file (see last line of paragraph 32). Hence, the compilation directives are used in the generation of the configuration file. But in claim 1, the first set of templates and the configuration data are used as inputs in the generation of a testbench. Paragraph 32 in Burgun makes clear that the configuration file is the output of the processing and not an input. Accordingly, the compilation directives being referred to in paragraph 32 are entirely different to the first set of templates being referred to in claim 1.

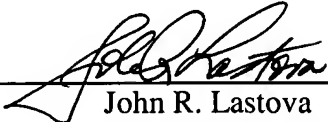
As identified above, there are a number of features missing from independent claims 1, 23-25, and 46. Accordingly, the anticipation rejection based on Burgun should be withdrawn.

The application is in condition for allowance. An early notice to that effect is earnestly solicited.

HOULIHANE  
Appl. No. 10/743,473  
August 18, 2005

Respectfully submitted,

**NIXON & VANDERHYE P.C.**

By:   
John R. Lastova  
Reg. No. 33,149

JRL:sd  
901 North Glebe Road, 11th Floor  
Arlington, VA 22203-1808  
Telephone: (703) 816-4000  
Facsimile: (703) 816-4100